Postel

RFC # 524
NIC # 17140

RECEIVED
JUN 22 1973

James E. White (JEW)
SRI-ARC
13 June 73

A Proposed Mail Protocol

AUTHOR'S INTENT

This is the document I offered in (15146,) to write. It's a proposed specification for handling mail in the Network -- a Mail Protocol.

Mail handling is currently implemented as two FTP commands, MAIL and MLFL, which permit an FTP user process to deliver a file or string of text to an FTP server process, designating it as mail to be made available to a user, identified by local name, in its host. The protocol proposed here is much richer than that, both in terms of the functions it supports, and in terms of the flexibility it provides.

Although one can (I think) and might, implement software on the basis of this document, this REALLY IS a Request for Comments. Comments, questions, position papers are solicited. There are, I'm sure, bugs in the protocol specified here, and I hope that readers will point them out via RFC as they discover them.

Various members of the Network community have, during the last few months, pointed out to me specific inadequacies in the existing mail commands and asked me to be conscious of them in designing a new protocol. I've tried to do that. If anyone feels that his concern wasn't properly dealt with here, or that it slipped through the cracks entirely (for which I apologize in advance), I would appreciate it if he would prod me once more.

INTRODUCTION 3

THE MAIL PROTOCOL ENVIRONMENT 3a

The Mail Protocol (MP) is implemented by Mail user and server processes, in keeping with the model for previous, high-level protocols. The Mail user and server processes are further specified to be also FTP user and server processes, respectively. That is, MP is implemented as a set of commands accessible from within the FTP command space. 3a1

The MP command set is defined to lie conceptually within a subsystem, invoked from the FTP command space with the command MAIL <CRLF>. 3a2

NOTE: SInce a command called 'MAIL' already exists within the FTP command space, the command name 'XMAIL' might substitute for 'MAIL' while the current mail commands are being phased out. 3a2a

The MP command set may or may not (according to the implementation of a particular server) be implemented by a process distinct from that which implements FTP proper. 3a3

The following are implications of the 'subsystem' concept, of which the reader (and implementer) must be aware: 3a4

(1) Names of MP commands are known only within the MP subsystem. MP commands must (and should naturally) be rejected by the server if the user process presents them outside of the subsystem. 3a4a

(2) Exit from the Mail subsystem (to the FTP command space) is effected with and only with the command EXIT <CRLF>. FTP commands must be rejected by the server if the user presents them while inside the subsystem (i.e., before EXIT is issued). 3a4b

(3) The same command name may be assigned without ambiguity to two entirely different commands, provided that one lies within the FTP command space and the other within MP, the two being distinguishable by their contexts. MP and FTP therefore do not compete for command names, and MP command names may be chosen without regard for the environment in which the subsystem resides. 3a4c

NOTE: It so happens that there are commands DEFINED

INTRODUCTION

within MP which duplicate the functions of FTP commands and bear the same names. The effective result is that some FTP commands are explicitly allowed within MP. The reader will understand that this fact is consistent with the conventions described in 1-3 above, and that no ambiguities result. 3a4d

The subsystem concept (if not the name 'subsystem') is taken from Mike Padlipsky's Unified User Level Protocol (UULP), which the author of the present document strongly supports. The fact that MP is accessed from FTP, rather than both FTP and MP being accessed independently from a more general executive program, is simply a concession to the fact that FTP is widely implemented and UULP isn't. The author hopes that protocol development will, in the near future, begin to proceed along the lines exemplified by UULP. 3a5

MP conforms to FTP in general syntax. In particular, commands and their responses are strings of NVT characters; command names are limited to four or fewer, upper- or lower-case, alphameric characters, and are terminated by the character SP; commands are generally terminated with the TELNET New Line sequence (CR LF); command responses contain both numeric (process readable) and text (human readable) portions. Both reader and implementer are referred to the FTP protocol document for a detailed description of such matters; no attempt has been made to duplicate the discussion in the present document. 3a6

The FTP protocol document assigns those replies whose second digit is '6' to RJE functions. In like manner, MP appropriates those reply codes whose second digit is '7' for reporting results peculiar to its functions. It is, however, the author's position that FTP, MP, and the RJE protocol are all best implemented as subsystems under a common UULP executive, in which case a single such subset of the reply space could be used unambiguously by all three protocols (and any yet to be defined), since every reply would implicitly be qualified by the name of the subsystem from which it emanates. 3a7

within MP which duplicate the functions of FTP commands and bear the same names. The effective result is that some FTP commands are explicitly allowed within MP. The reader will understand that this fact is consistent with the conventions described in 1-3 above, and that no ambiguities result.

The subsystem concept (if not the name 'subsystem') is taken from Mike Padlipsky's Untitled User Level Protocol (UULP), which the author of the present document strongly supports. The fact that MP is accessed from FTP, rather than both FTP and MP being accessed independently from a more general executive program, is simply a concession to the fact that FTP is widely implemented and UULP isn't. The author hopes that protocol development will, in the near future, begin to proceed along the lines exemplified by UULP.

MP conforms to FTP in general syntax. In particular, commands and their responses are strings of NVT characters; command names are limited to four or fewer, upper- or lower-case, alphanumeric characters, and are terminated with the TELNET New Line sequence (CR LF); command responses contain both numeric (process readable) and text (human readable) portions. Both reader and implementer are referred to the FTP protocol document for a detailed description of such matters; no attempt has been made to duplicate the discussion in the present document.

The FTP protocol document assigns those replies whose second digit is '5' to AUP functions. In like manner, MP appropriates those reply codes whose second digit is '7' for reporting results peculiar to its functions. It is, however, the author's position that FTP, MP, and the AUP protocol are all best implemented as subsystems under a common UULP executive, in which case a single such subset of the reply space could be used unambiguously by all three protocols (and any yet to be defined), since every reply would implicitly be qualified by the name of the subsystem from which it emanates.

THE MAIL MODEL 3b

MP defines mail to be text communicated between users (both human and processes) in less than (but ideally approaching) real time. The definition excludes so-called console-to-console communication, where users exchange information at the character or line level. 3b1

Pieces of mail are characterized by such attributes as title, content, author, and recipient. A piece of mail may be a one- or two-line message sent from one individual to another, a draft of a document sent by one individual to a design group for review, a polished, formal document sent from one group to another, a message sent to a human user by a process (e.g., an RJE server process might notify a user by Network mail when his job has completed), etc. All such forms of communication are mail and are supported by MP. 3b2

Pieces of mail can be forwarded from one location to another. 3b3

Pieces of mail can be replied to. 3b4

The identity of the author of a piece of mail can be verified, avoiding forgery and misrepresentation. 3b5

Pieces of mail can be permanently recorded, assigned a long-term identifer by which they can forever be retrieved for reference, and entered in catalogs. And access to such recorded mail can be restricted to a specified subset of the user community. 3b6

Some hosts accept mail whose recipients reside elsewhere in the Network, and assume responsibility for delivering the mail to them (worrying about retrying delivery when hosts are down, etc.), and acknowledging its delivery to the sender. 3b7

The picture being painted for the reader is one in which processes cooperate in various ways to flexibly move and manage Network mail. The author claims (without proof, of course) that the picture will in the future get yet more complicated, but that the protocol specified here can be conveniently enlarged to handle that picture, too. 3b8

ORGANIZATION OF THIS DOCUMENT 3c

The rest of this document consists of the following components: 3c1

GLOSSARY 3c1a

The concepts introduced briefly in the section above are more formally defined, and their manner of representation in the Protocol specified. 3c1a1

MP FUNCTIONS 3c1b

The command sequence is defined by which a user process initiates each of the logical functions (e.g., Distribution, Recording, Delivery) which can be performed by a Mail server process. 3c1b1

EXAMPLE 3c1c

An example of the command-response exchange between a user and server is given. 3c1c1

COMMAND SUMMARY 3c1d

A summary of MP commands is given. 3c1d1

COMMAND REPLIES 3c1e

Reply code assignments are given and briefly explained. 3c1e1

FORMAL SYNTAX 3c1f

The formal syntax of the command language is specified. 3c1f1

In all sections but the last (i.e., the formal syntax presentation), verbose keyword forms are employed, in the interests of clarity. These verbose forms have no existence anywhere but in this document; in implementing a Mail user or server process, the terse keyword forms which appear in the formal syntax presentation are to be employed. 3c2

ORGANIZATION OF THIS DOCUMENT

The text of this document consists of the following components:

GLOSSARY

The concepts introduced briefly in the section above are more formally defined, and their manner of implementation in the Protocol specified.

MP FUNCTIONS

The command sequence is defined by which a user process initiates each of the logical functions (e.g., distribution, Recording, Delivery) which can be performed by a Mail server process.

EXAMPLE

An example of the command-response exchange between a user and server is given.

COMMAND SUMMARY

A summary of MP commands is given.

COMMAND REPLIES

Reply code assignments are given and briefly explained.

FORMAL SYNTAX

The formal syntax of the command language is specified.

In all sections but the last (i.e., the formal syntax presentation), verbose keyword forms are employed, in the interests of clarity. These verbose forms have no existence anywhere but in this document; in implementing a Mail user or server process, the terse keyword forms which appear in the formal syntax presentation are to be employed.

GLOSSARY 4

Terms are listed here in alphabetical order. Words or phrases which appear in the definitions with initial letters capitalized are themselves formally defined elsewhere in the glossary. 4a

ACCESS LIST (for a piece of Recorded Mail) 4b

That set of Individuals with access to a piece of Recorded Mail, and for each such Individual, the type(s) of access granted to him. 4b1

An Access List is represented in the Protocol as a series of command pairs (juxtaposed in the command stream), each pair consisting of an ACCESS command followed immediately (and optionally) by an ACCESSTYPES command. Each pair of commands corresponds to one Individual in the set. 4b2

ACCESS <individual> <CA> 4b2a

ACCESSTYPES <accesstypes> <CA> 4b2b

Command arguments identify the Individual to whom access is granted, and specify the kind(s) of access allowed him. Either Read Access, Controlling Access, or both may be granted. 4b2b1

If no Individual is specified, All is implied. In the absence of an explicit ACCESSTYPES command, one with only Read Access specified is to be assumed. 4b2b2

In the absence of an explicit Access List, one granting Read Access to All and Controlling Access to the Author(s) and the Clerk is to be assumed. 4b2c

ACKNOWLEDGMENT (for a piece of Mail) 4c

A form of Unrecorded Mail, generated by a Distribution Agent, whose Recipient is the Monitor for a previous piece of Mail, which acknowledges Delivery -- successful or otherwise -- to the Recipient(s) of that first piece of Mail. 4c1

An Acknowledgment bears the Serial Number of the Mail it acknowledges, as the Reference Serial Number. 4c2

ACKNOWLEDGMENT CONDITION (for Acknowledgments) 4d

The attribute of an Acknowledgment which determines the circumstances under which it will be generated by the Distribution Agent. 4d1

The following Acknowledgment Conditions are defined: 4d2

ALWAYS 4d2a

Acknowledgment is given when all Deliveries are complete, regardless of whether or not they are all completed successfully. 4d2a1

FAILURE 4d2b

Acknowledgment is given when all Deliveries are complete if and only if Delivery to one or more Recipient(s) fails. 4d2b1

NEVER 4d2c

An Acknowledgment is never made. 4d2c1

An Acknowledgment Condition is represented in the Protocol by the command: 4d3

ACKCONDITION <ackcondition> <CA> 4d3a

In the absence of an explicit ACKCONDITION command, one with an argument of FAILURE is to be assumed. 4d4

ACKNOWLEDGMENT TYPE (for Acknowledgments and Progress Reports) 4e

The attribute of an Acknowledgment or Progress Report which determines the nature of its Content. 4e1

The following Acknowledgment Types are defined: 4e2

TERSE 4e2a

The Content of a TERSE Acknowledgment or Progress Report is specified by the Protocol to be an unembellished list of the Mail's Recipient(s), and the current Delivery Status for each (except that those Recipient(s) whose Delivery Status is SUCCESSFUL shall not be included in the list). 4e2a1

The Content of a TERSE Acknowledgment is one or more instances of the following: 4e2a2

<deliverystatus> <individual> <CRLF> 4e2a2a

TERSE Acknowledgments and Progress Reports are intended to be process-readable. 4e2a3

VERBOSE 4e2b

The Content of a VERBOSE Acknowledgment or Progress Report is not specified by the Protocol, but might include a list of those Recipient(s) to whom the Mail could not be Delivered and why, the times at which Delivery was made to others, etc. 4e2b1

VERBOSE Acknowledgments and Progress Reports are intended to be human-readable. 4e2b2

An Acknowledgment Type is represented in the Protocol by the command: 4e3

ACKTYPE <acktype> <CA> 4e3a

In the absence of an explicit ACKTYPE command, one with an argument of TERSE is to be assumed. 4e4

ALL 4f

Every conceivable Individual. 4f1

AUTHOR (of a piece of Mail) 4g

An Individual (there may be more than one) given formal recognition for having authored a piece of Mail. 4g1

AUTHOR LIST (for a piece of Mail) 4h

That set of Individuals who are Author(s) of a piece of Mail. 4h1

An Author List is represented in the Protocol as an Individual List of type AUTHOR. 4h2

CATALOG (of Recorded Mail) 4i

A named data base containing the Citation for each member of a set of logically related pieces of Recorded Mail. 4i1

CATALOG LIST (for a piece of Recorded Mail) 4j

CATALOG LIST (for a piece of recorded MAIL)

A named data base containing the CATALOG for each member
of a set of logically related pieces of recorded MAIL.

CATALOG (of recorded MAIL)

An AMPOT list is represented in the protocol as an
ordinary list of type CATALOG.

MAIL.
Data set of ingredients and the AMPOT(s) of a piece of

CONTENTS LIST (for a piece of MAIL)

Encoding for header-multiplexed a piece of MAIL.
An INGREDIENT (there may be more than one) given format

CONTENTS (of a piece of MAIL)

Easily concatenable INGREDIENT.

MAIL

That set of Catalogs which each contain the Citation for a piece of Recorded Mail. 4j1

A Catalog List is represented in the Protocol as a series of instances (juxtaposed in the command stream) of the following command. Each instance corresponds to one Catalog in the set. 4j2

CATALOG <catalog> <CA> 4j2a

CITATION (for a piece of Recorded Mail) 4k

The Static and Dynamic Attributes of a piece of Recorded Mail. 4k1

CITATION COMPONENT 4l

Any one of the Static or Dynamic Attributes which comprise a Citation. 4l1

CITATION RETRIEVAL (for a piece of Recorded Mail) 4m

The act of retrieving selected Citation Component(s). 4m1

CITATION TEMPLATE 4n

A specified subset of the Citation Component(s) for a piece of Recorded Mail. 4n1

A Citation Template is represented in the Protocol by the command: 4n2

CITATIONTEMPLATE <citaticntemp> <CA> 4n2a

The argument is a list of Citation Component(s). In the absence of an explicit CITATIONTEMPLATE command (or if the argument is null), one specifying Content only is to be assumed. 4n3

CLERK 4o

That Individual who prepares a piece of Mail for Recording, Distribution, or Delivery. Conceptually, the Individual with proof-reading responsibility for the piece of Mail. 4o1

A Clerk is represented in the Protocol as an Individual List of type CLERK and length 1. 4o2

A Proposed Mail Protocol

That set of Catalog which each contain the Citation for a piece of Recorded Mail.

A Catalog list is represented in the Protocol as a series of instances (juxtaposed in the command stream) of the following command. Each instance corresponds to one Catalog in the set.

CATALOG <catalog> <CA>

CITATION (for a piece of Recorded Mail)

The Static and Dynamic Attributes of a piece of Recorded Mail.

CITATION COMPONENT

Any one of the Static or Dynamic Attributes which comprise a Citation.

CITATION RETRIEVAL (for a piece of Recorded Mail)

The act of retrieving a selected Citation Component(s).

CITATION TEMPLATE

A specified subset of the Citation Component(s) for a piece of Recorded Mail.

A Citation Template is represented in the Protocol by the command:

CITATIONTEMPLATE <citationtemp> <CA>

The argument is a list of Citation Component(s). In the absence of an explicit CITATIONTEMPLATE command (or if the argument is null), one specifying Content only is to be assumed.

CLERK

That Individual who prepares a piece of Mail for Recording, Distribution, or Delivery. Conceptually, the Individual with proof-reading responsibility for the piece of Mail.

A Clerk is represented in the Protocol as an Individual list of type CLERK and length 1.

COMMENTS (for a piece of Mail) 4p

An informal, perhaps verbose description of the Content of a piece of Mail, or any other information the Author(s) wish to have made accessible to Recipient(s) of the Mail. 4p1

Comments are represented in the Protocol by the command: 4p2

COMMENTS <comments> <CA2> 4p2a

In the absence of an explicit COMMENTS command, one with a null argument is to be assumed. 4p3

CONTENT (of a piece of Mail) 4q

It's text. 4q1

Content is represented in the Protocol by either of the two commands below: 4q2

FILE <CA> 4q2a

The FILE command implies that the Content of the Mail will be transmitted (immediately) as a file using the FTP data transfer commands (e.g., BYTE, SOCK, TYPE) currently in effect. FILE is EXACTLY equivalent in use to an FTP STOR command (in its use of data transfer commands, in its establishment of the data connection, etc.), except that no pathname is required, and the server, rather than depositing the transmitted file in his file system, disposes of it in a manner appropriate for Mail. 4q2a1

TEXT <string> <CA2> 4q2b

The TEXT command implies that the Content of the Mail follows on the TELNET connection as a series of lines, each delimited from the preceeding one by CR LF, and terminated finally by CA2. 4q2b1

CONTROLLING ACCESS (to a piece of Recorded Mail) 4r

The right of an Individual to modify a Dynamic Attribute of a piece of Recorded Mail. 4r1

Recording Agents permit an Individual to modify a Dynamic Attribute of a piece of Recorded Mail if and only if he can properly identify himself as someone having Controlling Access to that Mail. 4r2

COMMENTS (for a piece of Mail)

An informal, perhaps verbose description of the Content of a piece of Mail, or any other information the Author(s) wish to have made accessible to Recipient(s) of the Mail.

Comments are represented in the Protocol by the command:

COMMENTS <comments> <CRLF>

In the absence of an explicit COMMENTS command, one with a null argument is to be assumed.

CONTENT (of a piece of Mail)

It's text.

Content is represented in the protocol by either of the two commands below:

FILE <CRLF>

The FILE command implies that the Content of the Mail will be transmitted (immediately) as a file using the FTP data transfer commands (e.g., BYTE, SOCK, TYPE) currently in effect. FILE is EXACTLY equivalent in use to an FTP STOR command (in its use of data transfer commands, in its establishment of the data connection, etc.), except that no pathname is required, and the server, rather than depositing the transmitted file in his file system, disposes of it in a manner appropriate for Mail.

TEXT <string> <CRLF>

The TEXT command implies that the Content of the Mail follows on the TELNET connection as a series of lines, each delimited from the preceding one by CR LF, and terminated finally by CRLF.

CONTROLLING ACCESS (to a piece of Recorded Mail)

The right of an Individual to modify a Dynamic Attribute of a piece of Recorded Mail.

Recording Agents permit an Individual to modify a Dynamic Attribute of a piece of Recorded Mail if and only if he can properly identify himself as someone having Controlling Access to that Mail.

CREATION DATE (of a piece of Mail) 4s

The date and time at which the final draft of a piece of Mail is completed by the Clerk before he releases it to a Delivery, Distribution, or Recording Agent for further processing. A single Creation Date is associated with each piece of Mail. In general, this date is different from the Delivery or Recording Date, since Mail often must be queued (for another host to come up) within the Delivery, Distribution, or Recording Agent's host before Delivery or Recording can occur. 4s1

A Creation Date is represented in the Protocol by the command: 4s2

CREATIONDATE <datetime> <CA> 4s2a

CUTOFF INTERVAL (for Distribution of a piece of Mail) 4t

That period of time, measured from the Distribute Date, after which the Distribution Agent is to abandon Delivery attempts for those Recipient(s) to whom Delivery has not yet been effected. 4t1

A Cutoff Interval is represented in the Protocol by the command: 4t2

CUTOFF <interval> <CA> 4t2a

In the absence of an explicit CUTOFF command, one specifying an interval of 7 days is to be assumed. 4t3

DELIVERY (of a piece of Mail) 4u

The act of transmitting a piece of Mail to the host of one of it's Recipient(s). 4u1

DELIVERY AGENT 4v

A process which effects Delivery of a piece of Mail. A Distribution Agent is by nature also a Delivery Agent. 4v1

DELIVERY DATE (of a piece of Mail to one of its Recipient(s)) 4w

The date and time at which a piece of Mail is Delivered by the Delivery Agent to a Recipient's system. A multitute of Delivery Dates, one for each Recipient, are associated with each piece of Mail. 4w1

CREATION DATE (of a piece of Mail)

The date and time at which the final draft of a piece of Mail is completed by the Clerk before he releases it to a Delivery, Distribution, or Recording Agent for further processing. A single Creation Date is associated with each piece of Mail. In general, this date is different from the Delivery or Recording Date, since Mail often must be queued (for another host to come up) within the Delivery, Distribution, or Recording Agent's host before Delivery or Recording can occur.

A Creation Date is represented in the Protocol by the command:

CREATIONDATE <date&time> <CA>

CUTOFF INTERVAL (for Distribution of a piece of Mail)

That period of time, measured from the Distribute Date, after which the Distribution Agent is to abandon Delivery attempts for those Recipient(s) to whom Delivery has not yet been effected.

A Cutoff Interval is represented in the Protocol by the command:

CUTOFF <interval> <CA>

In the absence of an explicit CUTOFF command, one specifying an interval of 7 days is to be assumed.

DELIVERY (of a piece of Mail)

The act of transmitting a piece of Mail to the host of one of it's Recipient(s).

DELIVERY AGENT

A process which effects Delivery of a piece of Mail. A Distribution Agent is by nature also a Delivery Agent.

DELIVERY DATE (of a piece of Mail to one of its Recipient(s))

The date and time at which a piece of Mail is Delivered by the Delivery Agent to a Recipient's system. A multitude of Delivery Dates, one for each Recipient, are associated with each piece of Mail.

DELIVERY STATUS (for a piece of Mail with respect to a Recipient) 4x

A measure of the extent to which a Delivery Agent has been successful, at a given point in time, in Delivering a piece of Mail to one of its Recipient(s). A multitute of Delivery Status', one for each Recipient, are associated with each piece of Mail. 4x1

The following Delivery Status' are defined: 4x2

FAILED 4x2a

Delivery was rejected by the Recipient's system (e.g., the connection request was rejected, the Mail server process doesn't support Delivery, the Recipient was not recognized by the server). 4x2a1

SUCCESSFUL 4x2b

Delivery was accomplished successfully. 4x2b1

TIMED OUT 4x2c

Either the Recipient's host was disconnected from the Net at every Delivery attempt, or no Mail server process has ever responded to the connection attempt. Hope of Delivery has been abandoned. 4x2c1

WAITING 4x2d

Either the Recipient's host has been disconnected from the Net at every Delivery attempt, or no Mail server process has yet responded to the connection attempt. Delivery attempts are continuing periodically. 4x2d1

UNATTEMPTED 4x2e

No Delivery attempt has yet been made. 4x2e1

DELIVERY TYPE (for a Delivery) 4y

The nature of the piece of Mail being delivered. 4y1

The following Delivery Types are defined: 4y2

FORWARD 4y2a

A Delivery of type FORWARD represents a piece of Recorded or Unrecorded Mail which is being Forwarded. 4y2a1

MAIL 4y2b

A Delivery of type MAIL represents a piece of Recorded or Unrecorded Mail whose ultimate source is an Individual. This is the "normal" Delivery Type. 4y2b1

NEGATIVE ACKNOWLEDGMENT 4y2c

A Delivery of type NEGATIVE ACKNOWLEDGMENT represents a piece of Unrecorded Mail generated by a Distribution Agent and acknowledging unsuccessful or partially unsuccessful Delivery of a previous piece of Mail (identified by Reference Serial Number) to it's Recipient(s). The Recipient for this piece of "system" Mail is the Monitor for the previous piece of Mail. 4y2c1

POSITIVE ACKNOWLEDGMENT 4y2d

A Delivery of type POSITIVE ACKNOWLEDGMENT represents a piece of Unrecorded Mail generated by a Distribution Agent and acknowledging successful Delivery of a previous piece of Mail (identified by Reference Serial Number) to it's Recipient(s) . The Recipient for this piece of "system" Mail is the Monitor for the previous piece of Mail. 4y2d1

PROGRESS REPORT 4y2e

A Delivery of type PROGRESS REPORT represents a piece of Unrecorded Mail generated by a Distribution Agent and reporting the Delivery Status for each Recipient of a previous piece of Mail (identified by Reference Serial Number). The Recipient for this piece of "system" Mail is the Monitor for the previous piece of Mail. 4y2e1

REPLY 4y2f

A Delivery of type REPLY represents a piece of Recorded or Unrecorded Mail generated in reply (or pertaining) to a previous piece of Mail (identified by Reference Serial Number). 4y2f1

A delivery of type BRIEF represents a piece of
a recorded or unrecorded MAIL delivered in reply (or
pertaining) to a previous piece of MAIL (identified
by reference serial number).

BRIEF

A delivery of type PROGRESS REPORT represents a
piece of unrecorded MAIL delivered by a distribution
agent and reporting its delivery status for each
piece of MAIL (identified by reference serial number).
Recipient of a previous piece of MAIL (identified by
reference serial number). The recipient for this
piece of "system" MAIL is the monitor for the
previous piece of MAIL.

PROGRESS REPORT

A delivery of type PROVISIONAL ACKNOWLEDGEMENT
represents a piece of unrecorded MAIL delivered by a
distribution agent and acknowledged successful or
partially unsuccessful delivery of a previous piece
of MAIL (identified by reference serial number) to
recipient(s). The recipient for this piece of
"system" MAIL is the monitor for the previous piece
of MAIL.

PROVISIONAL ACKNOWLEDGEMENT

A delivery of type MAIL represents a piece of
recorded or unrecorded MAIL which originated outside
the system.

MAIL

A delivery of type RECORDED represents a piece of
recorded or unrecorded MAIL which is being
forwarded.

Delivery Type is represented in the Prctocol by the command: 4y3

DELIVERYTYPE <deliverytype> <CA> 4y3a

In the absence of an explicit DELIVERYTYPE command, one with an argument of MAIL is to be assumed. 4y4

DISTRIBUTE DATE (for a piece of Mail) 4z

The date and time at which a piece of Mail is presented to a Distribution Agent for Distribution. 4z1

DISTRIBUTION (of a piece of Mail) 4a@

The act of Delivering a piece of Mail to its Recipient(s). Distribution can be effected by either the Clerk's Delivery Agent, or by a Distribution Agent acting on his behalf. 4a@1

DISTRIBUTION AGENT 4aa

A Mail server process which acts as intermediary in the delivery process by accepting Mail for Recipient(s) in hosts other than its own, and then assuming responsibility for Delivering the Mail to them and returning Acknowledgment to the appcinted Mcnitor. 4aa1

DISTRIBUTION LIST (for a piece of Mail) 4ab

In the Delivery or Distribution of a piece of Mail, that set of Individuals who are to receive Delivery of the Mail. 4ab1

In the Recording of Mail, that set of Individuals who have received formal and authorized Delivery of a piece of Mail. The list is kept current by Distribution Agents. Of course, any Individual with Read Access to the Mail can himself Deliver it informally to anyone he chooses without that fact's being reflected in the Distribution List. 4ab2

A Distribution List is represented in the Protoccl as a series of command quintuplets (juxtaposed in the command stream), each quintuplet consisting of a RECIPIENT command, followed immediately (and optionally) by any or all of the following in the order given: a GENERALDELIVERY, a GREETING, a SIGNATURE, and a DISPOSITION command. 4ab3

Each quintuplet corresponds to one Individual in the set. 4ab4

RECIPIENT <individual> <CA> 4ab4a

GENERALDELIVERY <CA> 4ab4b

This command is appropriate only in the context of the Delivery function. If the previous RECIPIENT command illicits the reply: 4ab4b1

474 Recipient unrecognized; is General Delivery OK? 4ab4b1a

the issuance of the GENERALDELIVERY command constitutes consent to proceed with General Delivery to that Recipient. If no such consent is given, the RECIPIENT command stands rejected. Unsolicited (i.e., unprompted for) GENERALDELIVERY commands in the Distribution List are treated by the server as NOPs. 4ab4b2

GREETING <greeting> <CA> 4ab4c

The GREETING command specifies the Greeting to be seen by the Recipient. 4ab4c1

If the first quintuplet in the list contains no GREETING command, one with a null argument is assumed. Thereafter, in the absence of an explicit GREETING command, one identical to that for the previous quintuplet is assumed. 4ab4c2

SIGNATURE <signature> <CA> 4ab4d

The SIGNATURE command specifies the Signature to be seen by the Recipient. 4ab4d1

If the first quintuplet in the list contains no SIGNATURE command, one with a null argument is assumed. Thereafter, in the absence of an explicit SIGNATURE command, one identical to that for the previous quintuplet is assumed. 4ab4d2

DISPOSITION <disposition> <CA> 4ab4e

The DISPOSITION command identifies the intent with which the Mail is Delivered to the Recipient by the Author(s), and may take any, all, or none of the following as arguments: 4ab4e1

Each quintuplet corresponds to one Individual in the set.

DESCRIBE <individual> <CA>

GENERALDELIVERY <CA>

This command is appropriate only in the context of the Delivery function. If the previous DESCRIBE command illicits the reply:

a2x Recipient unrecognized; is General Delivery OK?

the issuance of the GENERALDELIVERY command constitutes consent to proceed with General Delivery to that Recipient. If no such consent is given, the RECIPIENT command stands rejected. Unsolicited (i.e., uncompleted for) GENERALDELIVERY commands in the Distribution List are treated by the server as NOPs.

GREETING <greeting> <CA>

The GREETING command specifies the Greeting to be seen by the Recipient.

If the first quintuplet in the list contains no GREETING command, one with a null argument is assumed. Thereafter, in the absence of an explicit GREETING command, one identical to that for the previous quintuplet is assumed.

SIGNATURE <signature> <CA>

The SIGNATURE command specifies the Signature to be seen by the Recipient.

If the first quintuplet in the list contains no SIGNATURE command, one with a null argument is assumed. Thereafter, in the absence of an explicit SIGNATURE command, one identical to that for the previous quintuplet is assumed.

DISPOSITION <disposition> <CA>

The DISPOSITION command identifies the intent with which the Mail is Delivered to the Recipient by the Author(s), and may take any, all, or none of the following as arguments:

RSVP 4ab4e1a

The Author(s) request a Reply from the Recipient. 4ab4e1a1

ACTION 4ab4e1b

The Author(s) expect some action on the part of the Recipient. If ACTION doesn't appear, then the Mail is intended for the Recipient's information only. 4ab4e1b1

INTERRUPT 4ab4e1c

The Author(s) declare that examination of the Mail by the Recipient is urgent. In such cases, the Recipient's Mail server process may, upon Delivery, choose to interrupt the Recipient if he happens to be logged in at a terminal. 4ab4e1c1

No specific action in response to the presence of any of these arguments is required; the server is free if he likes to treat DISPOSITION commands as NOPs. 4ab4e2

The absence of a DISPOSITION command implies one with no arguments (i.e., for the Recipient's information only, no Reply required, and not urgent). 4ab4e3

DYNAMIC ATTRIBUTES (of a piece of Recorded Mail) 4ac

Those attributes of a piece of Recorded Mail -- Distribution List, Access List, and Catalog List -- which, though given initial values at Recording Time, can always be modified by an Individual with Controlling Access to the piece of Mail. 4ac1

FORWARDING (of Mail received for an Individual) 4ad

The act of transferring that set of Mail which has been previously Delivered to but not Read by an Individual, to another Individual. 4ad1

Users who are known at more than one host can cause their unRead Mail to be gathered in to a central location by performing the Forwarding function at each such host (both Individuals being, in this case, instances of the same

RSVP

The Author(s) request a Reply from the Recipient.

ACTION

The Author(s) expect some action on the part of the Recipient. It ACTION Doesn't appear, then the Mail is Intended for the Recipient's information only.

INTERRUPT

The Author(s) declare that examination of the Mail by the Recipient is urgent. In such cases, the Recipient's Mail server process may, upon Delivery, choose to interrupt the Recipient if he happens to be logged in at a terminal.

No specific action in response to the presence of any of these arguments is required; the server is free if he likes to treat DISPOSITION commands as NOPs.

The absence of a DISPOSITION command implies one with no arguments (i.e., for the Recipient's information only, no Reply required, and not urgent).

DYNAMIC ATTRIBUTES (of a piece of Recorded Mail)

Those attributes of a piece of Recorded Mail -- Distribution List, Access List, and Catalog List -- which, though given initial values at Recording Time, can always be modified by an Individual with Controlling Access to the piece of Mail.

FORWARDING (of Mail received for an Individual)

The act of transferring that set of Mail which has been previously Delivered to but not Read by an Individual, to another Individual.

Users who are known at more than one host can cause their unread Mail to be gathered in to a central location by performing the Forwarding function at each such host (both Individuals being, in this case, instances of the same

User). Mail which has been Forwarded is considered to have been Read at its source. 4ad2

FORWARDEE 4ae

That Individual whose Delivered but unRead Mail is to be Forwarded. 4ae1

A Forwardee is represented in the Protocol as an Individual List of type FORWARDEE and length 1. 4ae2

GENERAL DELIVERY (of a piece of Mail to an unrecognized Recipient) 4af

The act on the part of a server of accepting Delivery of a piece of Mail on behalf of an intended Recipient whose name the server doesn't recognize. The server retains the Recipient's name, and makes it and the other information provided by the user process available to some competent person, who attempts to make sense of the Recipient's name. If the Recipient is recognized, the Mail is 'hand' delivered to the appropriate Individual. 4af1

General Delivery of a piece of Mail to one of its intended Recipient(s) is performed only after the server informs the user process of its intent and receives the user process' consent. If that consent is not given, or if the server doesn't implement General Delivery, the server rejects the Delivery attempt for that Recipient. 4af2

Consent for General Delivery is represented in the Protocol by the command: 4af3

GENERALDELIVERY <CA> 4af3a

GREETING (for the Delivery of a piece of Mail to a Recipient) 4ag

A short greeting to a Recipient of a piece of Mail. 'Dear Dave' is a valid and perhaps typical Greeting. 4ag1

A Greeting is represented in the Protocol by the command: 4ag2

GREETING <greeting> <CA> 4ag2a

ID (for an Individual) 4ah

The password which an Individual may have to present to a Mail server process, to prove his identity. 4ah1

An Id is represented in the Protocol by the command: 4ah2

ID <id> <CA> 4ah2a

Ids have nothing to do with accounting, and when required by a server, they're required only to protect that server from forgery or misrepresentation. 4ah3

INDIVIDUAL 4ai

An instance of a User, identified by NIC Ident, or by the combination of host and Mailbox Name. 4ai1

INDIVIDUAL LIST (of type "t" and length "n") 4aj

A set of Individuals. 4aj1

An Individual List is represented in the Protocol as a series of "n" command pairs (juxtaposed in the command stream), each pair consisting of a "t" command, followed immediately by an ID command. Each pair corresponds to one Individual in the set. 4aj2

The ID command is given by the Mail user process at the option of the Mail server process; and whenever the server requires it, he must prompt for it with an appropriate reply to the preceeding "t" command. If no such prompt is given, the user process is not obliged to provide the ID command, but may if it chooses, in which case the server is obliged to treat it as if it had been prompted for and found correct. 4aj3

The ID command is a mechanism by which the server can assure himself that the user process is not acting without proper authorization from the Individual(s) involved, i.e., a mechanism by which a server can protect himself against forgery, misrepresentation, etc. 4aj4

"t" <individual> <CA> 4aj4a

ID <id> <CA> 4aj4b

MAIL 4ak

A body of text communicated from one set of Individual(s) to another, in less than (but ideally approaching) real time. 4ak1

MAILBOX NAME 4al

An Id is composed in the Protocol by the command: #ah2

ID <id> <CA> #ah2a

Ids have nothing to do with accounting, and when required by a server, they're required only to protect that server from forgery or misrepresentation. #ah3

INDIVIDUAL #ai

An instance of a User, identified by NIC Idno, or by the combination of host and mailbox Name. #ai1

INDIVIDUAL LIST (of type "i" and length "n") #aj

A set of Individuals. #aj1

An Individual List is represented in the Protocol as a series of "n" command pairs (juxtaposed in the command stream), each pair consisting of a "i" command, followed immediately by an ID command. Each pair corresponds to one Individual in the set. #aj2

The ID command is given by the Mail user process at the option of the Mail server process; and whenever the server requires it, he must prompt for it with an appropriate reply to the preceding "i" command. If no such prompt is given, the user process is not obliged to provide the ID command, but may if it chooses, in which case the server is obliged to treat it as if it had been prompted for and found correct. #aj3

The ID command is a mechanism by which the server can assure himself that the user process is not acting without proper authorization from the Individual(s) involved, i.e., a mechanism by which a server can protect himself against forgery, misrepresentation, etc. #aj4

"i" <individual> <CA> #aj4a

ID <id> <CA> #aj4b

MAIL #ak

A body of text communicated from one set of Individual(s) to another, in less than (but ideally approaching) real time. #ak1

MAILBOX NAME #al

The name a User employs at a host to send and receive Mail. 4a11

MONITOR (for a piece of Mail) 4am

The Individual who is the Recipient for Acknowledgments and Progress Reports. 4am1

A Monitor is represented in the Protocol as an Individual List of type MONITOR and length 1. 4am2

Monitor defaults to the Clerk if not explicitly specified. 4am3

PROGRESS REPORT (for a piece of Mail) 4an

A form of Unrecorded Mail, generated periodically during the distribution process by a Distribution Agent, whose Recipient is the Monitor for a previous piece of Mail, and whose Content is a list of the Recipient(s) and the current Delivery Status for each. A Progress Report bears the Serial Number of the Mail whose status it reports, as the Reference Serial Number. 4an1

PROTOCOL 4ao

The Mail Protocol (MP). 4ao1

READ (a piece of previously-Delivered Mail) 4ap

The act, on the part of a User, of examining a piece of Delivered Mail. 4ap1

READ ACCESS (to a piece of Recorded Mail) 4aq

The right of an Individual to retrieve the Content of a piece of Recorded Mail. 4aq1

Recording Agents permit an Individual to retrieve the Content of a piece of Recorded Mail if and only if he can properly identify himself as someone having Read Access to that Mail. An Individual can retrieve the Citation (except Content) from the Recording Agent independently of whether or not he has Read Access to the Mail. 4aq2

READ DATE (of a piece of Mail for one of its Recipient(s)) 4ar

The date and time, necessarily following Delivery, at which a piece of Mail is Read by a Recipient. A multitute

of Read Dates, one for each Recipient, are associated with each piece of Mail. 4ar1

RECIPIENT (of a piece of Mail) 4as

An Individual who has or is to receive Delivery of a piece of Mail. 4as1

RECORDED MAIL 4at

A piece of Mail whose Citation is available on a long-term (indefinite) basis from a Recording Agent. 4at1

RECORDING 4au

The service provided by a Recording Agent. 4au1

RECORDING AGENT 4av

A Mail server process which accepts Mail, permanently Records its Citation, and assigns a pathname by which that information can at any time be retrieved by an Individual with appropriate access. 4av1

RECORDING DATE (for a piece of Recorded Mail) 4aw

The date and time at which a piece of Mail is presented to a Recording Agent for Recording. A single Recording Date is associated with each piece of Recorded Mail. 4aw1

REFERENCE SERIAL NUMBER (for an Acknowledgment, Progress Report, or Reply) 4ax

The Serial Number of the piece of Mail to which an Acknowledgment, Progress Report, or Reply refers. 4ax1

A Reference Serial Number is represented in the Protocol by the command: 4ax2

REFERENCESERIAL <serialnumber> <CA> 4ax2a

In the absence of an explicit REFERENCESERIAL command, no Serial Number is to be assumed. 4ax3

REPLY (to a previous piece of Mail) 4ay

A piece of Recorded or Unrecorded Mail whose Author(s) are Recipient(s) of a previous piece of Mail, and which

replies or pertains to that same piece of Mail and bears its Serial Number, as the Reference Serial Number. 4ay1

REPORT INTERVAL (for a Progress Report) 4az

The interval between Progress Reports. 4az1

A Report Interval is represented in the Protocol by the command: 4az2

REPORTINTERVAL <interval> <CA> 4az2a

In the absence of an explicit REPORTINTERVAL command, one with an argument whose value is effectively infinite is to be assumed (i.e., no Progress Reports are to be made). 4az3

REQUESTOR 4b@

The Individual on whose behalf a Mail user process connects to and interacts with a Mail server process. 4b@1

A Requestor is represented in the Protocol as an Individual List of type REQUESTOR and length 1. 4b@2

SERIAL NUMBER (for a piece of Mail) 4ba

A short-term identifier, assigned to a piece of Mail by the Clerk (or his system), which accompanies Acknowledgments, Progress Reports, and Replies, and is used to correlate the latter with the former. The lifetime of a Serial Number is conceptually from its assignment by the Clerk until the Delivery of the Recipient(s) Reply(s) to the Author(s) (or until their decision to send no Reply). 4ba1

A Serial Number is represented in the Protocol by the command: 4ba2

SERIAL <serialnumber> <CA> 4ba2a

In the absence of an explicit SERIAL command, no Serial Number is to be assumed. 4ba3

SIGNATURE (for the Delivery of a piece of Mail to a Recipient) 4bb

A human-readable indication of the Author(s) of a piece of Mail. The string 'Jim and Dick' is a valid Signature. 4bb1

A Signature is respresented in the Protocol by the command: 4bb2

SIGNATURE <signature> <CA> 4bb2a

STATIC ATTRIBUTES (of a piece of Recorded Mail) 4bc

Those attributes of a piece of Recorded Mail -- Content, Title, Comments, Author(s), Clerk, and Creation Date -- which are forever fixed at Recording Time, and hence can never be modified. 4bc1

Static Attributes can be independently specified with commands described elsewhere, or specified collectively by reference to an existing piece of Recorded Mail. The command which follows assigns to the current piece of Mail the Static Attributes of the piece of Recorded Mail it references, and is exactly equivalent to an appropriate set of TITLE, COMMENTS, etc. commands. 4bc2

LOCATION <fileaddr> <CA> 4bc2a

TITLE (of a piece of Mail) 4bd

A concise description of the Content of a piece of Mail. 4bd1

A Title is represented in the Protocol by the command: 4bd2

TITLE <title> <CA> 4bd2a

In the absence of an explicit TITLE command, one with a null argument is to be assumed. 4bd3

UNRECORDED MAIL 4be

Mail which is never presented to a Recording Agent for permanent storage and catalcging, but which is simply Delivered to its Recipient(s) by a Delivery Agent. 4be1

UPDATE REQUEST (to a Recording Agent for a piece of Recorded Mail) 4bf

A request made of a Recording Agent to add, replace, or delete an Individual from the Access or Distribution List for a piece of Mail; or to add or delete a Catalog from the Catalog List. 4bf1

An Update Request is represented in the Protocol by the command: 4bf2

UPDATETYPE <updatetype> <CA> 4bf2a

followed immediately in the command stream by an Access, Distribution, or Catalog List. 4bf3

USER 4bg

A process or human who sends and/or receives Mail. 4bg1

USER VERIFICATION 4bh

The act of verifying an ID as that of a specified Individual. 4bh1

USER VERIFICATION AGENT 4bi

A Mail server process which performs User Verification. 4bi1

MP FUNCTIONS 5

A MP function is the request by a Mail user process and the subsequent performance by a server, of a major task related to the management of Mail. The following functions are defined: 5a

RECORDING
DELIVERY
DISTRIBUTION
FORWARDING
CITATION RETRIEVAL
UPDATE CITATION
USER VERIFICATION 5a1

One might expect that within the Network there would be just a few Recording Agents (who implement the Recording, Citation Retrieval, and Update Citation functions); a few Distribution Agents (who implement the Distribution function); one or two User Verification Agents (who implement the User Verification function); and many hosts who implement the Delivery and Forwarding functions. 5b

In general, a host is free to implement any, all, or none of the functions defined by the Protocol; and a host is free to require a login (for purposes of accounting) before permitting a user process access to any of the function(s) it has implemented. 5c

An FTP server process who chooses to not implement MP or a particular MP function simply rejects the command that requests the unimplemented server with the reply: 5d

400 Function not implemented. 5d1

A server who chooses to require login before allowing access to the MP subsystem or to an MP function, simply rejects the command that requests the charged-for service with the reply: 5e

332 Login first, please. 5e1

The functions defined in MP are: 5f

MP FUNCTIONS

A MP function is the request by a mail user process and the subsequent performance by a server, of a major task related to the management of Mail. The following functions are defined:

RECORDING
DELIVERY
DISTRIBUTION
FORWARDING
CITATION RETRIEVAL
UPDATE CITATION
USER VERIFICATION

One might expect that within the Network there would be just a few Recording Agents (who implement the Recording, Citation Retrieval, and Update Citation functions); a few Distribution Agents (who implement the Distribution function); one or two User Verification Agents (who implement the User Verification function); and many hosts who implement the Delivery and Forwarding functions.

To request a host in Europe to implement any all, or some of the functions defined by the Protocol; and a host is free to require a login (for purposes of accounting) before permitting a user process access to any of the function(s) it has implemented.

An FTP server process who chooses to not implement MP or a particular MP function simply rejects the command that requests the unimplemented server with the reply:

```
500 Function not implemented.
```

A server who chooses to require login before allowing access to the MP subsystem or to an MP function, simply rejects the command that requests the charged-for service with the reply:

```
332 Login first, please.
```

The functions defined in MP are:

RECORDING 5f1

The Recording function is invoked with the command: 5f1a

RECORD <CA> 5f1a1

Once this command is given, the user process shall provide the following (in any order that suits it): 5f1b

(1) Any Static Attributes desired. 5f1b1

Content and Clerk are required. Defaults for other Static Attributes (applied by the server if the appropriate commands don't appear) are as follows: 5f1b1a

Title or Comments as specified in the glossary. 5f1b1a1

Author to the Clerk. 5f1b1a2

Creation Date to the Recording Date. 5f1b1a3

(2) Initial values for any Dynamic Attributes desired. 5f1b2

Defaults (applied by the server if the appropriate commands don't appear) are as follows: 5f1b2a

Distribution and Catalog Lists to null. 5f1b2a1

Access List as specified in the glossary. 5f1b2a2

The Recording function is terminated with either of the commands: 5f1c

EXIT <CA> or ABORT <CA> 5f1c1

EXIT represents normal termination, and causes the server to perform the Recording function for which parameters have just been given. ABORT represents abnormal termination and effects exit from the function with no action having been taken by the server; the whole command exchange, beginning with RECORD, is therefore a NOP. 5f1d

DELIVERY 5f2

The Delivery function is invoked with the command: 5f2a

DELIVER <CA> 5f2a1

Once this command is given, the user process shall provide the following (in any order that suits it): 5f2b

(1) Any Static Attributes desired. 5f2b1

Content is required. Defaults for other Static Attributes (applied by the server if the appropriate commands don't appear) are as follows: 5f2b1a

Title or Comments as specified in the glossary. 5f2b1a1

Clerk to null. 5f2b1a2

Author to the Clerk. 5f2b1a3

Creation Date to the Delivery Date. 5f2b1a4

(2) Any Dynamic Attributes desired. 5f2b2

Distribution List is required. Defaults (applied by the server if the appropriate commands don't appear) are as follows: 5f2b2a

Catalog list to null. 5f2b2a1

Access List as specified in the glossary. 5f2b2a2

Both of these attributes are for the Recipient's information only when presented in the context of Delivery, so defaulting them to null simply implies that the Clerk doesn't desire that they be communicated to the Recipient. 5f2b2a2a

(3) Any or all of the following optional parameters: 5f2b3

(a) Delivery Type 5f2b3a

(b) Acknowledgment Type 5f2b3b

The specification of this parameter is

appropriate if and only if the Delivery Type is POSITIVE or NEGATIVE ACKNOWLEDGMENT or PROGRESS REPORT. In this context, Acknowledgment Type tells the server how to interpret the Content of the Acknowledgment. 5f2b3b1

(c) Serial Number 5f2b3c

The Serial Number assigned to the piece of Mail being Delivered. This parameter is inappropriate unless the Delivery Type is FORWARD (in which case the Serial Number is the one preserved from the previous Delivery), MAIL, or REPLY. 5f2b3c1

(d) Reference Serial Number 5f2b3d

The Serial Number assigned to the piece of Mail to which the current piece of Mail is either an Acknowledgment, Progress Report, or Reply. The specificaton of this parameter is therefore inappropriate if the Delivery Type is MAIL. 5f2b3d1

The Delivery function is terminated with either of the commands: 5f2c

EXIT <CA> or ABORT <CA> 5f2c1

EXIT represents normal termination, and causes the server to perform the Delivery function for which parameters have just been given. ABORT represents abnormal termination and effects exit from the function with no action having been taken by the server; the whole command exchange, beginning with DELIVER, is therefore a NOP. 5f2d

DISTRIBUTION

The Distribution function is invoked with the command:

DISTRIBUTE <CA>

Once this command is given, the user process shall provide the following (in any order that suits it):

(1) Any Static Attributes desired.

Content is required. Defaults for other Static Attributes (applied by the server if the appropriate commands don't appear) are as follows:

Title or Comments as specified in the glossary.

Clerk to null.

Author to the Clerk.

Creation Date to the Distribute Date.

(2) Any Dynamic Attributes desired.

Distribution List is required. Defaults (applied by the server if the appropriate commands don't appear) are as follows:

Catalog List to null.

Access List as specified in the glossary.

Both of these attributes are for the Recipient(s) information only when presented in the context of Distribution, so defaulting them to null simply implies that the Clerk doesn't desire that they be communicated to the Recipient(s).

(3) Any or all of the following optional parameters:

(a) Delivery Type

MAIL, FORWARD, or REPLY only.

(b) Serial Number

The Serial Number of the Mail being Distributed. The Distribution Agent will relay this Serial Number to each Recipient at Delivery. 5f3b3b1

(c) Reference Serial Number 5f3b3c

The Serial Number of the piece of Mail to which the current piece of Mail is a Reply. The Distribution Agent will relay this Serial Number to each Recipient at Delivery. The specification of this parameter is appropriate if and only if the Delivery Type is REPLY. 5f3b3c1

(d) Acknowledgment Condition 5f3b3d

An Acknowledgment is requested from the Distribution Agent if and only if the Acknowledgment Condition is other than NEVER. 5f3b3d1

(e) Acknowledgment Type 5f3b3e

(f) Cutoff Interval 5f3b3f

(g) Report Interval 5f3b3g

Progress Reports are requested from the Distribution Agent if and only if this parameter is specified explicitly. 5f3b3g1

(h) Monitor 5f3b3h

This parameter is ignored unless either an Acknowledgment or Progress Reports (or both) are requested. 5f3b3h1

The Distribution function is terminated with either of the commands: 5f3c

EXIT <CA> or ABORT <CA> 5f3c1

EXIT represents normal termination, and causes the server to perform the Distribution function for which parameters have just been given. ABORT represents abnormal termination and effects exit from the function with no action having been taken by the server; the whole command exchange, beginning with DISTRIBUTE, is therefore a NOP. 5f3d

FORWARDING 5f4

The Forwarding function is invoked with the command: 5f4a

FORWARD <CA> 5f4a1

Once this command is given, the user process shall provide the following (in any order that suits it): 5f4b

(1) Forwardee 5f4b1

(2) Distribution List 5f4b2

This is the set of Individual(s) to whom the Mail is to be Forwarded. 5f4b2a

The Forwarding function is terminated with either of the commands: 5f4c

EXIT <CA> or ABORT <CA> 5f4c1

EXIT represents normal termination, and causes the server to perform the Forwarding function for which parameters have just been given. ABORT represents abnormal termination and effects exit from the function with no action having been taken by the server; the whole command exchange, beginning with FORWARD, is therefore a NOP. 5f4d

CITATION RETRIEVAL 5f5

The Citation Retrieval function is invoked with the command: 5f5a

RETRIEVE <CA> 5f5a1

Once this command is given, the user process shall provide the following (in any order that suits it): 5f5b

(1) The pathname of the piece of Mail whose Citation is to be retrieved: 5f5b1

PATHNAME <pathname> <CA> 5f5b1a

(2) Any or all of the following optional parameters: 5f5b2

(a) Citation Template 5f5b2a

(b) Requestor 5f5b2b

This parameter is required if and only if Content is requested and Read Access happens not to be granted to All, in which case the server verifies that the Requestor has Read Access to the piece of Mail. 5f5b2b1

(c) FILE <CA> 5f5b2c

This command is appropriate if and only if Content is requested. The presence of this command implies that the Content of the Mail is to be returned to the user process (following the return on the TELNET connection of any other Citation Component(s) requested) as a file using the FTP data transfer commands (e.g., BYTE, SOCK, TYPE) currently in effect. FILE is exactly equivalent in effect to an FTP RETR command (in its use of data transfer commands, in its establishment of the data connection, etc.) except that no pathname is required. 5f5b2c1

In the absence of a FILE command, Content is returned on the TELNET connection like any other Citation Component. 5f5b2c2

The server returns the Citation Components in the order requested by the user process

(except that Content, if requested as a file, is always returned after the 'end of citation' indication), each as a reply whose numeric code is 172 and whose text is exactly the command normally used to communicate that same parameter to the server. A reply whose numeric code is 173 terminates the reply list. 5f5b2c3

Title and Content, which (in general) may each contain the TELNET New Line sequence (CR LF), are represented as continued replies, using the FTP reply continuation convention (see the FTP protocol document). The first four characters of each reply line except the first and last are blanks inserted by the server which must be deleted by the user process to correctly recover the value of the Title or Content. 5f5b2c4

The Citation Retrieval function is terminated with either of the commands: 5f5c

EXIT <CA> or ABORT <CA> 5f5c1

EXIT represents normal termination, and causes the server to perform the Citation Retrieval function for which parameters have just been given. ABORT represents abnormal termination and effects exit from the function with no action having been taken by the server; the whole command exchange, beginning with RETRIEVE, is therefore a NOP. 5f5d

UPDATE CITATION 5f6

The Update Citation function is invoked with the command: 5f6a

UPDATE <CA> 5f6a1

Once this command is given, the user process shall provide the following in the order specified: 5f6b

(1) Requestor 5f6b1

This parameter is required unless Controlling Access happens to be granted to All, in which case it is treatd as a NOP if given. The server verifies that the Requestor has controlling Access to the piece of Mail. 5f6b1a

(2) One or More Update Requests. 5f6b2

The Update Citation function is terminated with either of the commands: 5f6c

EXIT <CA> or ABORT <CA> 5f6c1

EXIT represents normal termination, and causes the server to perform the Update Citation function for which parameters have just been given. ABORT represents abnormal termination and effects exit from the function with no action having been taken by the server; the whole command exchange, beginning with UPDATE, is therefore a NOP. 5f6d

USER VERIFICATION 5f7

The User Verification function is invoked with the command: 5f7a

VERIFY <CA> 5f7a1

Once this command is given, the user process shall specify any number of Requestors. 5f7b

The server prompts for the Id for each, the user process provides it, and the server returns a reply whose numeric code is 272 if the ID is correct or 472 otherwise. 5f7b1

The User Verification function is terminated with either of the commands: 5f7c

EXIT <CA> or ABORT <CA> 5f7c1

EXAMPLE 6

In the example below, a short message is recorded for public access, and distributed to a single recipient. The user process is assumed already connected to the server. 6a

Note: This would be the implementation of NIC Journal Submission, where the NIC is understood to be both a Recording and Distribution Agent. 6a1

Replies from the server are in brackets. 6b

MAIL <CA> 6b1

The Mail subsystem is invoked.
[261 RE DE DI FW CI UP UV -- supported.] 6b1a

REC <CA> 6b2

The Recording function is invoked.
[200 OK.] 6b2a

TITL SMFS Runs on TENEX 1.31 at the NIC <CA> 6b3

A Title is given.
[200 OK.] 6b3a

TEXT The NIC came up on TENEX 1.31 on 1-APR. <CRLF>
I tried SMFS here on the new monitor and it <CRLF>
works fine. I don't understand why I had <CRLF>
problems running your copy of the code at <CRLF>
BBN-TENEX. Are you still unable to reference <CRLF>
the same archived file from two different <CRLF>
TENEXs? <CA2> 6b4

The Content of the message is entered.
[200 OK.] 6b4a

CLER WHITE@SRI-ARC <CR> 6b5

The Clerk is identified as White at SRI-ARC.
[330 OK. Now Id, please.] 6b5a

ID id <CA> 6b6

His Id is supplied.
[200 OK.] 6b6a

EXIT <CA> 6b7

EXAMPLE

In the example below, a short message is recorded for public access, and distributed to a single recipient. The user process is assumed already connected to the server.

Note: This would be the implementation of NIC Journal Submission, where the NIC is understood to be both recording and distribution agent.

Replies from the server are in brackets.

MAIL <CA>

The Mail subsystem is invoked.
[261 OK. Who dunnit?]

REC <CA>

The Recording function is invoked.
[200 OK.]

TITL SMFS Runs on TENEX 1.31 at the NIC <CA>

A Title is given.
[200 OK.]

TEXT The NIC came up on TENEX 1.31 on 1-APR. <CRLF>
I tried SMFS here on the new monitor and it <CRLF>
works fine. I don't understand why I had <CRLF>
problems running your copy of the code at <CRLF>
BBN-TENEX. Are you still unable to reference <CRLF>
the same archived file from two different <CRLF>
TENEXs? <CA2>

The Content of the message is entered.
[200 OK.]

CLER WHITE@SRI-ARC <CR>

The Clerk is identified as White at SRI-ARC.
[130 OK. Now Id, please.]

ID id <CA>

His Id is supplied.
[200 OK.]

EXIT <CA>

Exit from the Recording function is effected, and the pathname '15490' is returned by the Recording Agent for the now Recorded Mail.
[270 25490 -- is assigned as the pathname.] 6b7a

DIST <CA> 6b8

The Distribution functior is invoked.
[200 OK.] 6b8a

LOC SRI-ARC 15490 <CA> 6b9

The message just recorded is specified for Distribution.
[200 OK.] 6b9a

RECI * DHC <CA> 6b10

The Recipient is specified via NIC Ident tc be Dave Crocker at UCLA-NMC.
[200 OK.] 6b10a

GREE Dave <CA> 6b11

A greeting is given.
[200 OK.] 6b11a

DISP R <CA> 6b12

A reply is requested.
[200 OK.] 6b12a

SIGN Jim <CA> 6b13

The message is signed.
[200 OK.] 6b13a

ACKC A <CA> 6b14

Acknowledgment of the Mail's Delivery is requested whether Delivery succeeds or fails.
[200 OK.] 6b14a

ACKT T <CA> 6b15

The Acknowledgment is to be terse.
[200 OK.] 6b15a

CUT 1 D <CA> 6b16

Exit from the Recording function is effected, and the pathname "1549?0" is returned by the Recording Agent for the new Recorded Mail.
[270 25000 -- is assigned as the pathname.]

DIST <CA>

The Distribution Function is Invoked.
[200 OK.]

LOC SRI-ARC 15479 <CA>

The message just recorded is specified for Distribution.
[200 OK.]

RECI * DHC <CA>

The Recipient is specified via NIC ident to be Dave Crocker at UCLA-NMC.
[200 OK.]

GREE Dave <CA>

A greeting is given.
[200 OK.]

DISP R <CA>

A reply is requested.
[200 OK.]

SIGN Jim <CA>

The message is signed.
[200 OK.]

ACKC A <CA>

Acknowledgment of the Mail's Delivery is requested whether Delivery succeeds or fails.
[200 OK.]

ACKT T <CA>

The Acknowledgment is to be terse.
[200 OK.]

CUI 1 D <CA>

If Delivery hasn't been effected within 24 hours, the attempt is to be abandoned (and an Acknowledgment of failure returned). The Monitor (to whom the Acknowledgment is sent) is allowed to default to the Clerk.
[200 OK.] 6b16a

SERI serial <CA> 6b17

A Serial Number is assigned for purposes of coordinating Acknowledgment and Reply. A desireable implementation of the sender's user and server processes is one in which the Serial Number is assigned by the user process, rather than by the human user himself, in such a way that his server process can automatically make the association between original Mail, and subsequent Acknowledgment and Reply.
[200 OK.] 6b17a

EXIT <CA> 6b18

Exit from the Distribution function is effected.
[200 OK.] 6b18a

EXIT <CA> 6b19

Exit from the Mail subsystem is effected.
[200 OK.] 6b19a

COMMAND SUMMARY 7

Every command requires at least one reply from the server. 7a

THOSE SPECIFIC TO MP 7b

ABOFT <CA> 7b1
ACCESS <individual> <CA> 7b2
ACCESSTYPES <accesstypes> <CA> 7b3
ACKCONDITION <ackcondition> <CA> 7b4
ACKTYPE <acktype> <CA> 7b5
AUTHOR <individual> <CA> 7b6
CATALOG <catalog> <CA> 7b7
CITATIONTEMPLATE <citationtemp> <CA> 7b8
CLERK <individual> <CA> 7b9
COMMENTS <comments> <CA2> 7b10
CREATIONDATE <datetime> <CA> 7b11
CUTOFF <interval> <CA> 7b12
DELIVER <CA> 7b13
DELIVERYTYPE <deliverytype> <CA> 7b14
DISPOSITION <disposition> <CA> 7b15
DISTRIBUTE <CA> 7b16
EXIT <CA> 7b17
FILE <CA> 7b18
FORWARD <CA> 7b19
FORWARDEE <individual> <CA> 7b20
GENERALDELIVERY <CA> 7b21
GREETING <greeting> <CA> 7b22
ID <id> <CA> 7b23
LOCATION <fileaddr> <CA> 7b24
MAIL <CA> 7b25
MONITOR <individual> <CA> 7b26
PATHNAME <pathname> <CA> 7b27
RECIPIENT <individual> <CA> 7b28
RECORD <CA> 7b29
REFERENCESERIAL <serialnumber> <CA> 7b30
REPORTINTERVAL <interval> <CA> 7b31
REQUESTOR <individual> <CA> 7b32
RETRIEVE <CA> 7b33
SERIAL <serialnumber> <CA> 7b34
SIGNATURE <signature> <CA> 7b35
TEXT <string> <CA2> 7b36
TITLE <title> <CA> 7b37
UPDATE <CA> 7b38
UPDATETYPE <updatetype> <CA> 7b39
VERIFY <CA> 7b40

THOSE BORROWED FRCM FTP 7c

The following commands borrowed from FTP are defined (also) as MP commands to support the transfer of the Content of a piece of Mail in 'file' form. The reader is referred to the FTP protocol document for a description of their use and syntax. The borrowed commands are: 7c1

BYTE, SOCK, PASV, TYPE, STRU, MODE, REST, and SITE. 7c1a

The following commands borrowed from FTP are defined (also) as MP commands to permit changes of accounting parameters within the MP subsytem. The accounting parameters in force when the subsystem is entered apply (if necessary) within the subsystem until changed. Values to which the parameters may have been changed while in the subsystem continue in effect upon return to the FTP command space. The borrowed commands are: 7c2

USER, PASS, and ACCT. 7c2a

The following miscellaneous commands borrowed from FTP are defined (also) as MP commands: 7c3

HELP and NOOP. 7c3a

COMMAND REPLIES 8

This list is undoubtedly incomplete; some crucial reply code assignments may be missing despite the author's attempt to forsee the kinds of interaction that might arise between user and server and the responses from the server that they would require. 8a

172 <A citation Component> 8a1

In response to the EXIT command which terminates the Citation Retrieval function. 8a1a

173 End of citation. 8a2

Following a list of 172 replies. 8a2a

200 OK. 8a3

This is the standard, positive acknowledgment used throughout the Protocol. 8a3a

270 <pathname> -- is assigned as the pathname. 8a4

In response to the EXIT command which terminates the Record function. 8a4a

271 <functionlist> -- supported. 8a5

In response to the MAIL command by which the user process gains entry to the Mail subsystem. This response is mandatory, and from it the user process can quickly determine what function(s) are supported by the server. 8a5a

272 Requestor is who he says he is. 8a6

In response to an ID command in the User Verification function. This reply informs the user process that the Id given is in fact that of the Individual specified. 8a6a

330 OK. Now Id, please. 8a7

In response to the first command in each pair of commands in an Individual List. This reply requires the next command from the user process to be ID. 8a7a

332 Login first, please. 8a8

In response to any command which invokes a Mail function (e.g., RECORD, DISTRIBUTE, DELIVER), or to the MAIL command itself. This reply implies that the requested function is supported by the server, but that the user is required to login before invoking it. 8a8a

400 Function not implemented. 8a9

In response to any command which invokes a Mail function (e.g., RECORD, DISTRIBUTE, DELIVER), or to the MAIL command itself. This reply implies that the requested function is not supported by the server. 8a9a

431 Incorrect Id. 8a10

In response to the ID command in an Individual List command pair. This reply implies that the Id specified was incorrect. 8a10a

440 <Error relayed from Recording Agent> 8a11

In response to the LOCATION command. This reply implies that the server attempted to retrieve the specified piece of Mail from an FTP server but failed because it returned the error reply whose text is duplicated in the current reply. 8a11a

470 No such pathname. 8a12

In response to the PATHNAME command (in the Citation Retrieval function). This reply implies that the specified pathname is not recognized by the server. 8a12a

471 No unRead Mail to Forward. 8a13

In response to the EXIT command which terminates the Forwarding function. 8a13a

472 Requestor is NOT who he says he is. 8a14

In response to an ID command in the User Verification function. This reply informs the user process that the Id given is NOT that of the Individual specified. 8a14a

473 You don't have Read Access to the Mail. 8a15

In response to the LOCATION command, or to the PATHNAME command in a Citaton Retrieval function. This reply

In response to any command which invokes a Mail
function (e.g., RECORD, DISTRIBUTE, DELIVER), or to the
MAIL command itself. This reply implies that the
requested function is supported by the server, but that
the user is required to login before invoking it.

400 Function not implemented.

In response to any command which invokes a Mail
function (e.g., RECORD, DISTRIBUTE, DELIVER), or to the
MAIL command itself. This reply implies that the
requested function is not supported by the server.

411 Incorrect Id.

In response to the ID command in an Individual List
command pair. This reply implies that the Id specified
was incorrect.

440 <Error relayed from Recording Agent>

In response to the LOCATION command. This reply
implies that the server attempted to retrieve the
specified piece of Mail from an FTP server but failed
duplicated in the current reply.

470 No such pathname.

In response to the PATHNAME command (in the Citation
Retrieval function). This reply implies that the
specified pathname is not recognized by the server.

471 No unread Mail to Forward.

In response to the EXIT command which terminates the
Forwarding function.

472 Requestor is NOT who he says he is.

In response to an ID command in the User Verification
function. This reply informs the user process that the
Id given is NOT that of the Individual specified.

473 You don't have Read Access to the Mail.

In response to the LOCATION command, or to the PATHNAME
command in a Citation Retrieval function. This reply

implies that the Requestor doesn't have Read Access to the piece of Mail. 8a15a

474 Recipient unrecognized; is General Delivery OK? 8a16

In response to an instance of the RECIPIENT command in a Distribution List (in the context of the Delivery function). This response implies that the Recipient is unrecognized, but that the server will attempt General Delivery to him if the user process responds with a GENERALDELIVERY command; otherwise, the Recipient is rejected. 8a16a

475 That Individual is not at this host. 8a17

570 No such NIC Ident or Mailbox Name. 8a18

In response to any command in which a NIC Ident or Mailbox Name appears as an argument. This reply implies that the Individual specified does not exist. 8a18a

571 Invalid host. 8a19

In response to any command in which a host address or standard host name appears as an argument. This reply implies that no such host exists. 8a19a

572 No such catalog. 8a20

In respose to the CATALOG command. This reply implies that no such Catalog exists 8a20a

Any '500' reply. 8a21

Any of the error replies associated with FTP RETR/STOR commands. 8a22

Implies that the requestor doesn't have Read Access to the piece of Mail.

471 Recipient unrecognized; is General Delivery OK?

In response to an instance of the RECIPIENT command in a distribution List (in the context of the Delivery function). This response implies that the Recipient is unrecognized, but that the server will attempt General Delivery to him if the user process responds with a GENERALDELIVERY command; otherwise, the Recipient is rejected.

475 That Individual is not at this Host.

570 No such NIC Ident or Mailbox Name.

In response to any command in which a NIC Ident or Mailbox Name appears as an argument. This reply implies that the Individual specified does not exist.

571 Invalid Host.

In response to any command in which a host address or standard host name appears as an argument. This reply implies that no such host exists.

572 No such catalog.

In response to the CATALOG command. This reply implies that no such Catalog exists

Any '500' reply.

Any of the error replies associated with FTP RETR/STOR commands.

FORMAL SYNTAX 9

The terse keyword forms to be employed in actually implementing a Mail user or server process are generated by deleting character(s) from the corresponding verbose forms. Those deleted characters are included but enclosed in brackets throughout the description which follows. Spaces can be used freely between terminal elements of the syntax, and in some cases, at least one space must separate two elements whose boundary could not otherwise be distinguished. 9a

```
<CA2>              ::= TELNET Go Ahead character                          9a1
<CA>               ::= TELNET new line (CR LF)                            9a2
<CRLF>             ::= CR LF                                              9a3
<accesstypes>      ::= <readaccess> <controlaccess>                       9a4
<ackcondition>     ::= A[LWAYS] ! F[AILURE] ! N[EVER]                     9a5
<acktype>          ::= T[ERSE] ! V[ERBOSE]                                9a6
<action>           ::= A[CTION] ! null                                    9a7
<catalog>          ::= <string>                                           9a8
<citationccmp>     ::= D[ISTRIBUTION]L[IST] ! A[CCESS]L[IST]
                       ! C[ATALOG]L[IST] ! C[ON]T[ENT] !
                       T[ITLE] ! C[OM]M[ENTS] ! AU[THOR] !
                       CL[ERK] ! C[REATION]D[ATE]                         9a9
<citationtemp>     ::= <citationcomp> ! <citatonccmp>
                       <citationtemp>                                     9a10
<command>          ::= <shortbody> <CA> ! <longbody> <CA2>                9a11
<comments>         ::= <string>                                           9a12
<controlaccess>    ::= C[ONTROLLING] ! null                               9a13
<count>            ::= decimal integer                                    9a14
<date>             ::= <daycfmonth> / <month> / <year>                    9a15
<datetime>         ::= <date> <time>                                      9a16
<dayofmonth>       ::= decimal integer, 1-31                              9a17
<days>             ::= <count> D[AYS]                                     9a18
<deliverystatus>   ::= F[AILED] ! S[UCCESSFUL] ! T[IMED OUT]
                       ! W[AITING] ! U[NATTEMPTED]                       9a19
<deliverytype>     ::= F[ORWARD] ! M[AIL] ! N[EGATIVE
                       ACKNOWLEDGMENT] ! P[OSITIVE
                       ACKNOWLEDGMENT] ! P[ROGRESS]R[EPORT]
                       ! R[EPLY]                                          9a20
<disposition>      ::= <rsvp <action> <interrupt>                         9a21
<fileaddr>         ::= <host> <pathname>                                  9a22
<functionlist>     ::= <functiontype> ! <functiontype>
                       <functionlist>                                     9a23
<functiontype>     ::= RE[CORDING] ! DE[LIVERY] !
                       DI[STRIBUTION] ! F[OR]W[ARDING] !
                       CI[TATION RETRIEVAL] ! UP[DATE] !
                       U[SER]V[ERIFICATION]                               9a24
<globalname>       ::= * <nicident>                                       9a25
<greeting>         ::= <string>                                           9a26
```

A Proposed Mail Protocol

FORMAL SYNTAX

The terse keyword forms to be employed in actually implementing e-mail user or server process are generated by deleting character(s) from the corresponding verbose forms. Those deleted characters are included but enclosed in brackets throughout the description which follows. Spaces can be used freely between terminal elements of the syntax, and in some cases, at least one space must separate two elements whose boundary could not otherwise be distinguished.

```
<CA2>           ::= TELNET Go Ahead character
<CA>            ::= TELNET new line (CR LF)
<CRLF>          ::= CR LF
<accesstypes>   ::= <readaccess> <controlaccess>
<ackcondition>  ::= A[LWAYS] ! S[ATISFD] ! N[EVER]
<acktype>       ::= T[ERSE] ! V[ERBOSE]
<action>        ::= A[CTION] ! null
<catalog>       ::= <string>
<citationcomp>  ::= D[ISTRIBUTION][IST] ! A[CCESS][IST]
                    ! C[ATALOG][IST] ! C[ON]T[ENT] !
                    T[ITLE] ! C[OM]M[ENTS] ! AU[THOR] !
                    CL[ERK] ! C[REATION][AT]
                    <citationtemp>
<command>       ::= <shortbody> <CA> ! <longbody> <CA2>
<comments>      ::= <string>
<controlaccess> ::= C[ONTROLLING] ! null
<count>         ::= decimal integer
<date>          ::= <daycfmonth> \ <month> \ <year>
<datetime>      ::= <date> <time>
<daycfmonth>    ::= decimal integer, 1-31
<days>          ::= <count> D[AYS]
<deliverystatus> ::= F[AILED] ! S[UCCESSFUL] ! F[INED OUT]
                    ! W[AITING] ! U[NATTEMPTED]
<delivertype>   ::= F[ORWARD] ! M[AIL] ! N[EGATIVE
                    ACKNOWLEDGMENT] ! P[OSITIVE
                    ACKNOWLEDGMENT] ! P[ROGRESS][REPORT]
                    ! R[EPLY]
<disposition>   ::= <resp <action> <interrupt>
<fileaddr>      ::= <host> <pathname>
<functionlist>  ::= <functiontype> ! <functiontype>
                    <functionlist>
<functiontype>  ::= R[CONTING] ! DE[LIVERY] !
                    DI[STRIBUTION] ! F[OR]W[ARDING] !
                    CI[TATION RETRIEVAL] ! UP[DATE] !
                    U[SER]V[ERIFICATION]
<globalname>    ::= * <nicagent>
<greeting>      ::= <string>
```

```
<host>            ::= <hostname> ! <hostaddress>                          9a27
<hostaddress>     ::= decimal integer, 0-255                              9a28
<hostname>        ::= standard host name                                  9a29
<hour>            ::= decimal integer, 0-23                               9a30
<hours>           ::= <count> H[OURS]                                     9a31
<individual>      ::= <localname> ! <globalname>                          9a32
<interrupt>       ::= I[NTERRUPT] ! null                                  9a33
<interval>        ::= <days> ! <hours> ! <days> <hours>                   9a34
<localname>       ::= <mailbox> @ <host> ! <mailbox> @                    9a35
   NOTE:  Host defaults to that of the server.                            9a35a
<longbody>        ::= COM[MENTS] <comments> !
                      TEXT <string>                                       9a36
<mailbox>         ::= <string>                                            9a37
<minute>          ::= decimal integer, 0-59                               9a38
<month>           ::= decimal integer, 1-12                               9a39
<nicident>        ::= <string>                                            9a40
<id>              ::= <string>                                            9a41
<pathname>        ::= <string>                                            9a42
<readaccess>      ::= R[EAD] ! null                                       9a43
<rsvp>            ::= R[SVP] ! null                                       9a44
<serialnumber>    ::= <string>                                            9a45
<shortbody>       ::= ABOR[T] !                                           9a46
                      ACC[ESS] <individual> !                             9a46a
                      ACKC[ONDITION] <ackcondition> !                     9a46b
                      ACKT[YPE] <acktype> !                               9a46c
                      AC[CESS]TY[PES] <accesstypes> !                     9a46d
                      AUTH[OR] <individual> !                             9a46e
                      CAT[ALOG] <catalog> !                               9a46f
                      CLEB[K] <individual> !                              9a46g
                      CR[EATION]DA[TE] <datetime> !                       9a46h
                      CUT[OFF] <interval> !                               9a46i
                      C[ITATION]TEM[PLATE] <citationtemp> !               9a46j
                      DELI[VER] !                                         9a46k
                      DE[LIVERY]TY[PE] <deliverytype> !                   9a46l
                      DISP[OSITION] <dispositicn> !                       9a46m
                      DIST[RIBUTE] !                                      9a46n
                      EXIT !                                              9a46o
                      FILE !                                              9a46p
                      FOR[WARDE]E <individual> !                          9a46q
                      FOR[WARD] !                                         9a46r
                      GEN[ERAL]D[ELIVERY] !                               9a46s
                      GREE[TING] <greeting> !                             9a46t
                      ID <id> !                                           9a46u
                      LOC[ATION] <fileaddr> !                             9a46v
                      MAIL !                                              9a46w
                      MON[ITOR] <individual> !                            9a46x
                      PATH[NAME] <pathname> !                             9a46y
                      RECI[PIENT] <individual> !                          9a46z
                      REC[ORD] !                                          9a46a@
```

```
<boat>          ::= <boatname> ! <boataddress>
<boataddress>   ::= decimal integer, 0-2755
<boatname>      ::= standard boat name
<hour>          ::= decimal integer, 0-23
<hours>         ::= <count> BE[CUBS]
<individual>    ::= <localname> ! <globalname>
<interrupt>     ::= IE[INTERRUPT] ! null
<interval>      ::= <days> ! <hours> ! <days> <hours>
<localname>     ::= <mailbox> @ <host> ! <mailbox>
                NOTE:  Host defaults to that of the server.
<loudbody>      ::= CONF[MENTS] <comments> !
                    TEXT <string>
<mailbox>       ::= <string>
<minute>        ::= decimal integer, 0-59
<month>         ::= decimal integer, 1-12
<nickname>      ::= <string>
<id>            ::= <string>
<pathname>      ::= <string>
<readaccess>    ::= RE[FAD] ! null
<rsvp>          ::= RE[SVP] ! null
<serialnumber>  ::= <string>
<shortbody>     ::= ABORT[] !
                    ACCE[SS] <individual> !
                ACK[CONDITION] <ackcondition> !
                ACKT[YPE] <acktype> !
                ACC[CESS]TY[PES] <accesstypes> !
                AUTH[OR] <individual> !
                CAT[ALOG] <catalog> !
                CLEA[R] <individual> !
                CRE[ATION]DA[TE] <datetime> !
                CUT[OFF] <interval> !
                C[ITATION]TRE[PLATE] <citationtemp> !
                DELI[VER] !
                DEL[IVERY]TY[PE] <deliverytype> !
                DISP[OSITION] <disposition> !
                DIST[RIBUTE] !
                EXIT !
                FILE !
                FOR[WARDE]E <individual> !
                FOR[WARD] !
                GEN[ERAL]DE[LIVERY] !
                GREE[TING] <greeting> !
                ID <id> !
                LOC[ATION] <filedir> !
                MAIL !
                MON[ITOR] <individual> !
                PATH[NAME] <pathname> !
                REC[IPIENT] <individual> !
                REC[ORE] !
```

```
                    REQ[UESTO]R <individual> !                          9a46aa
                    RETR[IEVE] !                                        9a46ab
                    R[EFERENCE]SER[IAL] <serialnumber> !                9a46ac
                    R[EPORT]INT[ERVAL] <interval> !                     9a46ad
                    SERI[AL] <serialnumber> !                           9a46ae
                    SIGN[ATURE] <signature> !                           9a46af
                    TITL[E] <title> !                                   9a46ag
                    UPDA[TE] !                                          9a46ah
                    UP[DATE]TY[PE] <updatetype> !                       9a46ai
                    VER[IFY]                                            9a46aj
<signature>     ::= <string>                                            9a47
<string>        ::= any non-zero number of visible
                    characters
                    (in particular, CA and CA2 are
                    excluded)                                           9a48
<time>          ::= <hour> : <minute> <timezone>                        9a49
<timezone>      ::= EST ! EDT ! CST ! CDT ! MST ! MDT !
                    PST ! PDT ! GMT                                     9a50
<title>         ::= <string>                                            9a51
<updatetype>    ::= A[DD] ! R[EPLACE] ! D[ELETE]                        9a52
<year>          ::= full year in decimal (e.g., 1973)                   9a53
```